

CLAIMS

We claim:

1. A method of storing data pages at a proxy, the method comprising:
 - a) receiving a data page;
 - 5 b) receiving page dependency data that contains one or more dependencies such that each dependency indicates an underlying data source which the said data page is dependent on;
 - c) storing said data page; and
 - d) storing said page dependency data.
- 10 2. A method as defined in claim 1, where said page dependency data are written in HTML or XML.
3. A method as defined in claim 1, where said page dependency data are generated by a Request-Based dependency generator.
4. A method as defined in claim 3, where said Request-Based dependency generator
15 uses a URL request of the said data page.
5. A method as defined in claim 3, where said Request-Based dependency generator uses a configuration file to generate said page dependency data.
6. A method as defined in claim 1, where said page dependency data are generated by a script-based dependency generator.
- 20 7. A method as defined in claim 6, where said page dependency data are encoded in the said data page by a script-based dependency generator.

002198-061200

8. A method as defined in claim 1, where said page dependency data are manually encoded into a data file.
9. A method as defined in claim 1, where said data page and said page dependency data are stored in one or more files.
- 5 10. A method as defined in claim 1, where said storing of said data page at the proxy is in response to data in a configuration file.
11. A method as defined in claim 1, further comprising:
 - a) receiving an event;
 - b) determining if said event changes one of the said page dependency data associated
10 with said data page; and
 - c) updating the cache by refreshing or deleting said data page.
12. A method as defined in claim 11, where send event is received incorporated in an event message.
13. A method as defined in claim 11, where said event is written in HTML or XML.
- 15 14. A method as defined in claim 11, where said event came from a Request-Based event generator.
15. A method as defined in claim 14, where said Request-Based event generator uses a configuration file.
16. A method as defined in claim 14, where said Request-Based event generator uses a
20 URL request of the said data page.

002254 v04.PA (4C2604!.DOC)

17. A method as defined in claim 16, where said URL request is parsed to obtain parameters.
18. A method as defined in claim 16, where said URL request includes request header information.
- 5 19. A method as defined in claim 18, where said URL request is parsed to obtain parameters.
20. A method as defined in claim 11, where said event came from a script-based event generator.
21. A method as defined in claim 11, where said event came from a trigger-based event generator.
- 10 22. A method as defined in claim 11, where said event came from polling event generator.
23. A method as defined in claim 11, where said event came from a custom event generator.
- 15 24. A method as defined in claim 11, where said determination if said event changes one of the page dependency data associated with said data page is done by a change event evaluator.
25. A method as defined in claim 11, where said determination by the change event evaluator is done by matching said page dependency data with said event.
- 20 26. A method as defined in claim 11, where said event was generated by Request-Based event generator and sent to a change event evaluator.

27. A method as defined in claim 11, where said updating the cache involves keeping the index of URL addresses and page dependency data up-to-date.

28. A computer software product for use in a computer system that executes program steps recorded in a computer-readable media to perform a method for enabling storing of data pages at a proxy comprising:

a) a recordable media; and

b) a program of computer-readable instructions executable by the computer to perform method steps comprising:

i) receiving a data page;

ii) receiving page dependency data that contains one or more dependencies such that each dependency indicates an underlying data source which the said data page is dependent on;

iii) storing said data page;

iv) storing said page dependency data;

v) receiving an event;

vi) determining if said event changes said one of the page dependency data associated with said data page; and

vii) updating the cache by refreshing or deleting said data page.

29. A proxy server system that provides stored data files without requesting data files from the origin web server, comprising:

a) a central processing unit that can establish communication with a user computer;

- b) a storage device;
- c) a processor connected to the storage device wherein the storage device stores:
 - i) at least one program component for controlling the processor; and
- d) the processor is operative with said program component to:

- 5 i) receive a data page;
- ii) receive page dependency data that contains one or more dependencies
 such that each dependency indicates an underlying data source which the
 said data page is dependent on;
- iii) store said data page;
- 10 iv) store said page dependency data;
- v) receive an event;
- vi) determine if said event changes said one of the page dependency data
 associated with said data page; and
- vii) update the cache by refreshing or deleting said data page.

15

20